On the Cost-Quality Tradeoff in Topology-Aware Overlay Path Probing

Chiping Tang and Philip K. McKinley
Software Engineering and Network Systems Laboratory
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824
{tangchip,mckinley}@cse.msu.edu

Abstract

Path probing is essential to maintaining an efficient overlay network topology. However, the cost of a full-scale probing is as high as $O(n^2)$, which is prohibitive in large-scale overlay networks. Several methods have been proposed to reduce probing overhead, although at a cost in terms of probing completeness. In this paper, an orthogonal solution is proposed that trades probing overhead for estimation accuracy in sparse networks such as the Internet. The proposed solution uses network-level path composition information (for example, as provided by a topology server) to infer path quality without full-scale probing. The inference metrics include latency, loss rate and available bandwidth. This approach is used to design several probing algorithms, which are evaluated through analysis and simulation. The results show that the proposed method can significantly reduce probing overhead while providing bounded *quality estimations for* all $n \times (n-1)$ *overlay paths. The solution is well suited to medium-scale overlay networks in the Internet. In other environments, it can be combined with extant probing algorithms to further improve performance.*

1. Introduction

Overlay networks have recently attracted attention, due to their ability to support applications such as end-system multicast, structured peer-to-peer systems, global event notification, resilient routing, and denial-of-service prevention. Overlay applications exploit the high flexibility, deployability, and computing capability of end systems to provide better network services.

The choice of overlay network topology has significant effect on system performance. For example, a suboptimal multicast tree may result in high link stress and high RDP (relative delay penalty) [1], while a poor-quality backup path provides little fault tolerance. In order to construct a good overlay topology in dynamic environments such as the Internet, overlay nodes need to periodically probe the paths to other nodes and monitor the qualities of those paths.

One straightforward solution is pair-wise probing, which is employed by Narada [1] and RON [2]. Although this approach is complete and accurate, the number of probing packets, and therefore the probing overhead, is $O(n^2)$, where n is the number of overlay nodes. In a sparse network such as the Internet [3], overlay paths are likely to share physical links, and thus the pair-wise probing may incur high link stress. The worst-case link stress is $O(n^2)$, where it happens that a single "bridge" link connects two subnetworks that equally partition the overlay nodes. Although this is an extreme case, our simulation results show the worst-case link stress can still be considerably higher than $O(n)$ in real networks. For example, Figure 1 plots link stress for an AS-level Internet topology [4] (note both axes are logarithmic). High link stress not only affects the normal data traffic on that link, but also affects the probing results due to link sharing among overlay paths.

Several methods have been proposed to reduce probing overhead in overlay networks [5]. In structured peer-to-peer systems, for example, a node maintains connections to only $O(\log n)$ [6,7] or $O(n^{1/d})$ [8] neighbors. Each node periodically probes its neighbors and replaces poorly connected neighbors as appropriate. Thus the overall probing overhead is reduced to $O(n \log n)$. In scalable application-level multicast systems such as NICE [9] and HMTP [10], nodes are organized in a hierarchy based on their distances to each other. Each node periodically selects a particular set of nodes to probe, in order to find its optimal location in the hierarchy. Usually the size of the probing set is a constant, so the overall probing overhead is $O(n)$. Although the total number of probing packets, as well as worst case link stress, can be substantially reduced, these approaches sacrifice probing "completeness": only a small subset of the possible paths is probed in each round. Despite the effort to select promising paths to probe, it is still possible that a num-

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**

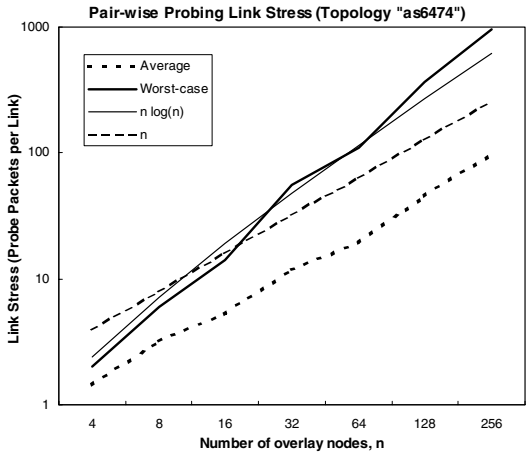


Figure 1. Link stress of pair-wise probing in an AS-level Internet topology.

ber of high-quality paths remain unknown for a long period of time, especially in dynamic networks.

In this paper, we propose a probing strategy that can significantly reduce probing overhead from $O(n^2)$ while providing a bounded quality estimation for $all\ n \times (n-1)$ paths. The method uses network topology information and exploits the overlap among overlay paths. The general approach is based on the following observations: (1) the latency of a path is greater than the latency of any of its subpaths; (2) the loss rate of a path is greater than the loss rate of any of its subpaths; and (3) the available bandwidth of a path is lower than the available or bottleneck bandwidth of any of its subpaths. After probing a path we make coarse quality estimation for all physical links on that path, from which we further infer the quality of the paths that contain these physical links.

Our proposed solution is most suitable for applications that require bounded, rather than precise, estimations for all paths. A typical example is RON [2]. In RON, each node periodically probes all other nodes to find all possible alternative paths. As an alternative, our proposed approach can provide a bounded quality estimation for every path, while incurring much lower probing overhead. If we only need to bypass path outages, the estimation bound can be very loose, where any path with finite latency will be eligible.

The performance of our proposed solution is dependent on the following assumptions: (1) paths between pairs of overlay nodes significantly overlap with one another; (2) route changes are less frequent than path quality changes; and (3) the network-level composition of every overlay path is known by at least one overlay node. Assumption (1) is self-evident in the Internet, where the average vertex degree is a constant [3]. Assumption (2) is generally true since a route change is usually caused by path quality change, while the reverse does not necessarily hold. Although assumption (3) has not generally held in the past, increasingly end node techniques such as *traceroute* and *topology servers* [11] are employed to obtain network topology information. Moreover, methods of inferring topology from end-to-end measurements [12–15] can be used. Other researchers studying overlay networks assume the availability of topology information at end nodes [16, 17], and we expect this trend to continue. In our future work we will study how incomplete or inaccurate topology information affects the performance of the proposed probing algorithms.

The contributions of our work are as follows. First, we propose a novel approach to solve the overlay path probing problem. Compared with existing approaches, this solution can increase probing completeness while maintaining comparable or lower probing overhead and provide reasonable estimation accuracy. Moreover, the solution is orthogonal to existing approaches and thus can be combined with them to further improve performance. Second, we designed a suite of probing algorithms that trade probing cost with estimation accuracy. We demonstrate the strengths and weaknesses of these algorithms in different situations and in terms of latency, loss rate, and available bandwidth. Third, we evaluated the performance of our solution through extensive simulation. Both generated and real Internet topologies are used in the performance study.

The remainder of the paper is organized as follows: In Section 2, we review background topics and introduce related works. In Section 3, we formally define the overlay path probing problem and analyze possible solutions. In Section 4, we introduce several algorithms for path selection. The tradeoff between probing cost and estimation accuracy is addressed. We describe the performance evaluation results in Section 5, and discuss application-related issues in Section 6. We make concluding remarks and discuss possible future research directions in Section 7.

2. Background and Related Work

Network dynamics may significantly affect the performance of distributed applications, and therefore robust and efficient distributed systems need to adapt their behavior to environmental changes. Since the performance of overlay networks is sensitive to changes in path quality, path probing is essential for maintaining efficient overlay networks. Systems targeting small overlay networks [1, 2] usually employ pair-wise probing. Larger overlay networks employ techniques such as hierarchy, approximation or aggregation to improve the probing scalability [5–7, 9, 10]. As with all these approaches, our solution aims to reduce probing overhead. However, we focus on the tradeoff between probing cost and probing accuracy, instead of on the tradeoff between probing cost and probing completeness.

One disadvantage of overlay networks is communication inefficiency caused by the disparity between the logical and physical topology. Although network topology information is typically not available at end nodes, many end-to-end approaches [12–15] can be employed to obtain, or infer, such information. Many overlay systems can benefit from topology information, and increasingly protocols are designed to exploit such information. For example, Kwon and Fahmy [16] proposed a protocol that builds efficient application-level multicast trees based on topology information. Cui, Stoica and Katz [17] proposed a correlated link failure probability model to allocate backup paths in overlay networks. Their approach implicitly exploits network topology information to construct the failure probability model. In this paper we explore how overlay network probing might benefit if such information is widely available.

Our proposed method probes a subset of overlay paths periodically, and estimates the quality of other paths from these probing results. This method is similar to network tomography approaches that infer link delay [18–20], loss [21–23], or available bandwidth [24] from end-to-end measurements. However, those methods aim at quality inference for *network links*, while we address end-to-end *path* quality. In addition, those approaches focus on inference accuracy, and therefore can employ computationally intensive statistical methods such as the EM algorithm [25] to infer quality metrics. Such algorithms are not practical for online quality estimation in large networks [26]. Instead, our methods use only simple arithmetic operations and comparisons, rendering them more efficient and practical for probing periods of several seconds. Naturally, the estimation is less accurate. However, the required level of accuracy in overlay applications is not as high as in network tomography applications. Moreover, we can always improve the accuracy by dispatching more probes as needed.

Barford *et al.* [27] addressed the problem of efficient network topology measurement. In their approach, adding more measurement sites will produce a larger observable topology. However, they show that the utility of adding additional sites declines quickly, which means that recently-added resources are less productive than the early-added resources from the perspective of node or link discovery. Similarly, we also find in our system that recent probe packets are less effective than the early packets from the perspective of accuracy improvement. We quantify the marginal utility in terms of estimation accuracy. Based on this observation, we propose to stop adding probe packets after a certain point so that the cost-quality ratio is minimized.

Bejerano and Rastogi [28] proposed a two-phase approach to minimize the cost and probe traffic in network monitoring. They compute the location of a minimal set of monitoring stations such that all network links can be monitored using probe messages from any of these stations. Their computation is based on greedy approximation algorithms to the set cover problem. In our solution, we also use approximation algorithms to set cover problems to find a minimum set of paths whose links subsume all constituent physical links. The paths in the minimum cover are probed, and the results are used to obtain bounded estimations for all other paths. In contrast to their work [28], which studies how to choose a set of nodes for robust and efficient link monitoring, we study how the selection of paths affects the estimation accuracy, assuming the set of nodes is known.

Shavitt *et al.* [29] defined the concept of path segment, and proposed an algorithm based on algebraic tools to compute additional path distances that are not explicitly measured. We adopt the path segment concept in this paper to describe our topology-aware overlay network. In addition, we use a similar approach to compute path latency and loss rate based on linear equations constructed from probing results. In their work, the set of paths that can be measured is a subset of all paths. Instead, we assume all paths can be measured, and thus focus on how to select paths to compute additional path distances (delay or loss rate).

3. The Problem and Our Approach

In this section we formally introduce the overlay path probing problem and our solution. First we present a formal definition for this problem and define evaluation metrics. Then we describe our general probing framework. Lastly we investigate the problem and propose possible solutions.

3.1. Problem Definition

An overlay network is a logical abstraction of the underlying physical network. As shown in Figure 2, we can represent an overlay network as a graph $G =< V, E >$, where the vertex set V is the collection of all overlay nodes, and the edge set E is the collection of paths between overlay node pairs that are determined by the application. The routers and other end nodes not involved in the application are abstracted away from the vertex set V, and the physical links of the selected overlay paths are not included in the edge set E. This abstraction substantially simplifies the network graph as well as related network algorithms, which is one of the appealing characteristics of overlay networks. We point out that while we actually model an overlay network as a directed graph, we use undirected graphs in examples to simplify the discussion.

Let $G_c =< V, E_c >$ be the complete graph derived from the vertex set V. The construction of an overlay network is equivalent to the selection of a subset E of E_c. To measure the quality of the overlay topology and the cost of

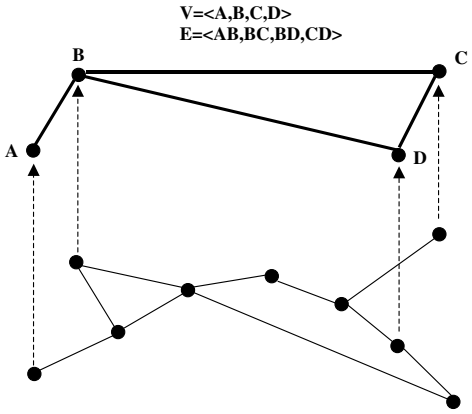


Figure 2. An overlay network.

the construction procedure, we introduce a numeric quality function $f(G)$ and attach three metrics Q, Q' and C, where $f(G)$ is the quality of overlay network G in terms of application-specific metrics, $Q(e)$ is the real quality of edge e, $Q'(e)$ is the observed (measured or inferred) quality of e, and $C(e)$ is the probing cost of e. All these metrics are non-negative real values.

Let $G' =< V, E' >$ be the overlay network constructed using the observed quality Q' of edge set E'. We assume that $f(G')$ is maximized when Q' is closest to Q, that is, when the measured or inferred quality is closest to the real quality. To measure the distance between Q' and Q, we introduce the metric *estimation error*, δ. We define δ to be the normalized error between a numeric value x and its estimation x', specifically, $\delta(x, x') = \frac{|x-x'|}{\max(x,x')}$. We note that the observed quality of e is determined by both its real quality and the set of edges $P \subseteq E_c$ chosen for probing, that is, $Q'(e) = h(P, Q(e))$. We define *estimation accuracy* as follows.

Definition 1: The *estimation accuracy of path p under probing set P* is $Z(p, P, Q) = 1 - \delta(Q(p), h(P, Q(p))) = 1 - \delta(Q(p), Q'(p))$.

Definition 2: The *overall estimation accuracy under probing set P* is, $Z(P, Q) = \sum_i w(i)Z(p_i, P, Q)$, a weighted sum of estimation accuracy of all paths.

3.2. General Approach

The approach we propose here to the overlay path-probing problem is centralized (our on-going work addresses distributed solutions [30]). A node is elected as a $leader$, and interacts with a topology server to maintain a minimal topology that covers all nodes and links involved in the overlay network. The leader is responsible for selecting paths for probing. It requests a node at one end of each path to probe the other end of the path. These nodes then send back the probing results to the leader, which generates bounded quality estimations for other paths. The approach trades probing overhead for estimation accuracy by adjusting the size of the probing set. The low computational overhead makes the approach suitable for online monitoring.

It seems the leader node is a potential performance bottleneck and a single point of failure. We argue, however, that the per-round computational cost at the leader node is low in small- and medium-scale networks. In addition, our approach adds only a small number of probing instruction packets if the leader is not required to disseminate the inference results. So the communication cost at the leader node does not increase much. On the other hand, the computation and communication cost at other nodes are greatly reduced when compared to the pair-wise probing approach.

For the concern of single point of failure, we assume that leader election and leader backup strategies can be used. Moreover, we point out that the leader model enables us to integrate our algorithms with hierarchical solutions, where a leader node is also selected for each cluster. For example, we can adopt our solution in a cluster where the topology data is available, the path overlapping degree is high, and the traffic is dynamic, while using the other hierarchical probing techniques in other clusters.

Finally, since our concern in this paper is on quality estimation and the cost-quality tradeoff, we do not address single path measurement. We can use any of several proposed end-to-end techniques and tools [31] to measure latency, loss rate, and bandwidth. We adopt a coarse time scale for these metrics and assume they are stationary within one probing cycle. Therefore we can combine different probing results in a cycle to infer the quality of other paths.

3.3. Problem Analysis

We now return to the problem analysis. Normally the observed quality of paths is defined only for those paths that have been probed. Formally,

$$h(P, Q(e)) = \begin{cases} Q(e), & \text{if} \quad \text{e} \in \text{P}; \\ undefined, & \text{otherwise} \end{cases} .$$

Although we might guess the quality of the unprobed paths, this estimation is not bounded and is likely harmful to the construction of an optimal overlay network. Therefore we define $\delta(x, undefined) = 1$.

As noted earlier, overlay paths often overlap. We can exploit this information to reduce probing overhead. We extend the overlay network model defined above to include *path segments*. A path segment is a subpath of a physical path (*i.e.* network-level path) and comprises one or more physical links. Every overlay path in turn comprises one or more path segments.

Definition 3: A *path segment* is one of the maximal subpaths in a path such that all the inner vertices on the subpath are not incident to any other physical links in the overlay network.

Our approach begins by converting the set of overlay paths to path segments. The result is a *topology-aware overlay network*, denoted $G_T =< V, S >$, where S is the path segment set. Figure 3 shows the topology-aware overlay network that corresponds to the overlay network in Figure 2.

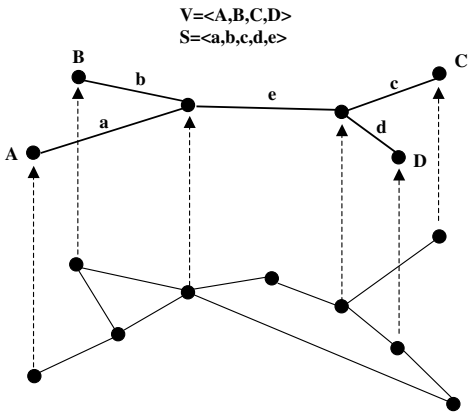


Figure 3. Topology-aware overlay network.

We construct the path segment set S as follows: for each overlay path, find its corresponding physical path. If the path overlaps with any segment in S, split the path into several subpaths so that the subpaths are either identical to one segment in S, or they are disjoint with all segments in S. If this is infeasible because a segment in S contains some of the subpaths but is not identical to them, then split those segments into subpaths. Repeat this process until any two subpaths or segment-subpath pair are either disjoint or identical. Add the newly generated subpaths to S as new segments. Discard any redundant segments. If a physical path does not overlap with any segment in S, add it to S as a single segment. Select the next overlay path and repeat this process. After the construction of path segment set S, every overlay path can be expressed as one or several path segments in S. The construction algorithm guarantees that each path segment in S is disjoint with all other segments.

The path segment set S reflects the overlapping relationship among overlay paths. We can exploit this data structure to produce bounded path quality estimation. We use a two-phase approach. First, we compute a quality estimation for every segment in all probed paths. Second, we obtain a quality estimation for all unprobed paths from the quality estimation of segments.

In the first phase, after probing a path, the quality of every segment in that path is bounded by the probing results. This method is based on the following observations: (1) the latency of a path is greater than the latency of any of its subpaths; (2) the loss rate of a path is greater than or equal to the loss rate of any of its subpaths; and (3) the available bandwidth of a path is less than or equal to the available bandwidth of any of its subpaths. For example, if the measured loss rate of path AB in Figure 3 is 5%, then the loss rate of neither segment a nor segment b can be higher than 5%. Thus, we can obtain a (coarse) bounded estimation for all path segments on that path. Since many path segments are shared among overlay paths, they may have different sources of quality bounds. For example, suppose the probing result for path AB constrains the loss rate of segment a to be no higher than 5%, while the result for AC indicates the loss rate of a is no higher than 3%. Since both constraints should be met, segment a obtains a more accurate loss rate upper bound of 3%. Similarly, we can obtain upper bounds for latency and lower bounds for available bandwidth for all segments in probed paths. The estimation accuracy increases as more relevant overlay paths are probed.

In the second phase, when we have obtained bounded quality estimation for all the segments in an unprobed path p from the probing of other paths, we can generate a bounded estimation for p. This time the estimation is based on slightly different observations: (1) the latency of a path is no greater than the sum of the latency upper bounds of all its segments; (2) the loss rate of a path p is no greater than $1 - \prod_{s \in p}(1 - r_s)$, where r_s is the loss rate upper bound of segment s; and (3) the available bandwidth of a path is no lower than the minimum of the available bandwidth lower bound of all its segments. For example, having probed paths AB and AC in Figure 3, we can make an assertion with high confidence that the loss rate of path BC is no greater than 11% $(1 - .95 \times .97 \times .97)$, even though BC was not probed. Obviously the path quality estimation is more accurate for shorter paths.

4. Algorithms

In this section we present several path selection and quality estimation algorithms. First, we describe a generic probing and estimation algorithm that realizes the approach described above. Then we explore the possibility of exploiting set cover and other techniques to select paths in order to reduce probing overhead and improve estimation accuracy. Lastly, we apply an algebraic method [29] to obtain more accurate latency and loss rate estimations.

4.1. Generic Algorithm

Our algorithm first selects a set of unprobed overlay paths and sends probe packets to these paths (path selec-

tion is discussed in Section 4.2). After obtaining the probing results, the algorithm updates the estimation of all segments in those paths. The update rule is as follows: for the latency and loss rate metrics, the algorithm assigns the probing result value of each path as a new estimation to the relevant segments whose previous estimation values are $greater$ than the new value. For the available bandwidth metric, it assigns the new value to those that have $lower$ previous estimation values.

After updating segment estimates, the algorithm re-evaluates the quality of all affected paths. For each updated segment, it identifies all *unprobed* paths that contain this particular segment. It then updates the quality estimation of these paths according to the following rule: for latency, the algorithm sums the estimation values of all segments in each path, and assigns this sum as the new path latency estimation. For loss rate, the algorithm re-calculates $1 - \prod_{s \in p}(1 - r_s)$, where r_s is the loss rate estimation of segment s, and assigns this value as the new path loss rate estimation. For available bandwidth, the algorithm checks if the updated estimation value of the particular segment is the $lowest$ among all segments in each of these paths. For paths where the answer is yes, the algorithm assigns this value as the new path bandwidth estimation.

4.2. Path Selection

The probing overhead can be calculated based on the number (or a weighted sum) of probing packets, the total physical link stress caused by probing traffic, or the worst-case link stress caused by probing traffic. The choice of the metric is application specific, and it will affect the optimality of our path selection algorithm. We design different path selection algorithms for different probing overhead metrics.

To maximize overall estimation accuracy, we adopt a two-phase strategy: (1) obtain bounded quality estimations for all paths (set cover phase); (2) select the most promising paths to probe that are most likely able to refine the estimation of other paths (direct selection phase).

In the set cover phase, we need to use probes to find a valid estimation for every path segment in S, from which we can compute a quality estimation for all paths. Covering every segment with least probing overhead is an instance of the minimum set cover problem [32]. In this case, each path is a set, and the segments in a path are the elements of the set. We consider three different strategies for applications with different probing overhead definitions:

(1) *no cover*. This is a degenerate case, where we actually skip the set cover phase. We eventually find a valid estimation for every path segment in the second phase.

(2) *minimum set cover*. If we consider the *number* of probing packets as the probing overhead, then this is the standard set cover problem. We can use a greedy heuristic [33, 34] to obtain the set cover that incurs minimum probing overhead. The basic idea is to choose, at each step, the path with maximum number of unprobed segments.

(3) *minimum weighted set cover*. If probing path e incurs cost $C(e)$, then the problem corresponds to the weighted set cover problem, and the proposed heuristic [34] is to choose the path with the minimum ratio of its weight to the number of its unprobed segments. If we consider the total link stress as the probing overhead, then this also corresponds to the weighted set cover problem, where the weight of path e is its physical hop count.

In the direct selection phase, the path selection strategy is different from that of the set cover phase. In the set cover phase, we focus on finding a minimum cover: a probing on already probed segments is considered a waste. In the direct selection phase, however, a probing on already probed segments can be used to refine the quality estimation of those segments. Based on this observation, we propose four path selection strategies for this phase:

(1) *random*. We randomly select paths to probe.

(2) *lowest cost*. Considering that the probing cost of a path is proportional to the number of its segments, and a path with fewer segments produces a more accurate estimation, a straightforward strategy is to always pick the unprobed path with the lowest cost.

(3) *least segments*. A variation of strategy (2) is to choose the path with the least number of segments.

(4) *most references*. The probing result of a path that overlaps with other paths will improve the estimation accuracy of those paths. Specifically, the minimization (or maximization) function used in upper bound (or lower bound) inference will return tighter bounds under more inputs. Therefore our fourth strategy is to choose the path with the maximum reference number, where we define the reference number of a path to be the number of other paths overlapping with it.

In summary, we propose three set cover strategies: no cover, minimum set cover, and minimum weighted set cover. We consider four direct selection strategies after set cover phase: random, lowest cost, least segments, and most references. In the lowest cost strategy we break ties by least segments. In all other strategies we break ties by probing cost. Combining these strategies produces 12 different algorithms, which are listed in Table 1. Intuitively no single algorithm will always be optimal. Rather, the algorithm should be selected based on the network topology and the application specifications.

path selection strategies	set cover strategies		
	none	minimum set cover (minimize number of probes)	minimum weighted set cover (minimize total probing cost $\sum_e C(e)$)
random	NULL_RANDOM	SETCOVER_RANDOM	WSETCOVER_RANDOM
lowest cost	NULL_MINCOST	SETCOVER_MINCOST	WSETCOVER_MINCOST
least segments	NULL_MINSEG	SETCOVER_MINSEG	WSETCOVER_MINSEG
most references	NULL_MAXREF	SETCOVER_MAXREF	WSETCOVER_MAXREF

Table 1. Path selection algorithms.

4.3. Improving Latency and Loss Rate Estimation

There exists a fundamental difference between latency estimation and bandwidth estimation. The estimation of a segment's available bandwidth may be equal to its real value. By probing more paths containing a particular segment, we will have higher chance of obtaining the real available bandwidth of the segment. However, the latency of a segment is calculated as the minimum latency of the paths that include this segment. In other words, the latency estimation of a segment is a sum (instead of maximum or minimum) of the segment's real latency and the latency of several other segments. Therefore, the estimation will never be equal to the real segment latency, unless there is a probed path that comprises only this segment.

To solve the problem, we employ the algebraic approach proposed by Shavitt *et al.* [29]. We exploit the same accumulative property of path latency. Let $L(e)$ be the latency estimation of path e or segment e. As an example, assume the measured latency of path AB in Figure 3 is a constant L_{AB}, the latency of AC is L_{AC}, and the latency of CD is L_{CD}. Then we have the following linear equations:

$$\begin{cases} L(a) + L(b) = L_{AB}, \\ L(a) + L(c) + L(e) = L_{AC}, \\ L(BC) = L(b) + L(c) + L(e), \\ L(AD) = L(a) + L(d) + L(e), \\ L(c) + L(d) = L_{CD}, \\ L(BD) = L(b) + L(d) + L(e) \end{cases}$$

There exists an infinite number of solutions for BC, AD, and BD that satisfy all these constraints. However, if we add the probing results of $L(BC)$ and $L(AD)$, then we can solve all the linear equations. Generally, if we can find $|S|$ paths to probe, and the resulting linear equations correspond to a $|S| \times |S|$ non-singular matrix, then we can solve all $|S|$ linear equations and obtain the real latency of all segments in $|S|$, from which we can derive the latency of all unprobed paths. The power of this approach is that, usually $|S| \ll n^2$, which implies that we can obtain accurate latency estimations for all $n \times (n-1)$ paths with relatively few probes.

In reality, the rank of the matrix may be less than $|S|$. In this case, we cannot solve for all the variables. However, our goal is to estimate quality of paths instead of segments. A rank-deficient matrix indicates that some equations can be expressed as linear combinations of other equations. In our system, this correponds to our being able to derive the quality of all paths from a small set of probing results. Our simulation results show the savings can be substantial. Moreover, we can use the same strategy on loss rate estimation. Assume the loss rate of path p is, $r_p = 1 - \prod_{s \in p}(1 - r_s)$, where s is a segment of path p. Let $R = \log(1 - r)$, then we can convert the loss rate observation to a linear equation: $R_p = \sum_{s \in p} R_s$.

To summarize, we estimate path latency and loss rate using the following procedure:

1. Select paths using our path selection algorithms.
2. If the segment vector of a late-selected path is linearly dependent on any segment vector of paths selected earlier, then discard this path and try other unselected paths.
3. Probe the selected paths.
4. Use the strategy in our generic algorithm to derive segment latency and loss rate estimations.
5. If we can solve any subset of the linear equations, then solve them and obtain exact latency values for a subset of segments.
6. For each unprobed path, if its latency (loss rate) value can be expressed as a linear combination of the quality value of some probed paths, then derive path latency (loss rate) from other paths. Else, compute path latency (loss rate) from segment quality values.

5. Performance Evaluation

In this section we present the performance evaluation results of our algorithms. First we introduce the simulation environments. Then we describe the probing cost for given levels of estimation accuracy. After that, we describe the estimation accuracy for any set of selected paths. Lastly, we compare the performance of different path selection algorithms, as well as the performance of the original and the improved algorithms.

5.1. Simulation Setup

We have evaluated our algorithms on six different physical network topologies, including a real AS-level Internet topology [4], three generated by the GT-ITM [35] topology generator, and two generated by Inet3.0 [36]. The performance of the algorithms differs slightly among the topologies. Generally, larger and denser topologies diffuse overlay paths and provide less space for the tradeoff. In this paper we report only the results on the "as6474" topology, which is the real Internet AS-level topology as of February 2000. The network has 6474 nodes with an average degree of 2.15. Please refer to [37] for the details of other topologies and the corresponding performance results.

We vary the size of the overlay network from 4 to 256. The overlay nodes are uniformly selected from the underlying physical topologies. We assume the IP layer uses shortest path routing with delay as the metric. In our path selection algorithms the weight of a path is set to the number of physical hops between two end nodes. We execute the probing algorithms on each configuration 10 times with different random seeds and report the average, or representative, results. The variance of data in different simulation runs is very small. Therefore we do not show confidence intervals in the figures for the sake of clarity.

We vary the delay for backbone links from 1ms to 50ms. The delay on links from edge routers to end hosts is randomly set between 1ms and 3ms. We use the LM1 model [21] for spatial distribution of backbone link loss rate, where the good link fraction, f, is set to 90%. The loss rate on a "good" link is between 0 and 1%, and the loss rate of a "bad" link is between 5% and 10%. For edge links, the good link fraction is set to 50%, the "good" link loss rate is between 0 and 1%, and the "bad" link loss rate is between 10% and 20%. We randomly set available bandwidth between 100MB to 500MB for backbone links, and 500KB to 1MB for edge links.

5.2. Probing Cost

Probing cost is proportional to estimation accuracy: the more accurate the estimation, the more probes required. In the extreme case, we need all $n \times (n-1)$ probes for a 100% accurate estimation. In this subsection we examine the probing cost for given levels of overall estimation accuracy. We set the estimation weights of all segments to be equal, so that the overall estimation accuracy is equivalent to the average estimation accuracy. The results shown in this subsection are for the SETCOVER_RANDOM algorithm. Some selected results of other algorithms are shown in a later subsection that compares algorithm performance.

We plot the number of needed probes versus network size for several estimation accuracy levels in Figure 4. Figure 4(a) is for latency estimation, Figure 4(b) is for loss rate estimation, and Figure 4(c) is for bandwidth estimation. The network size varies from 4 to 256. First we consider the minimum number of probes needed to produce bounded estimation for all paths. This corresponds to the size of a minimum set cover. The curves in Figure 4 marked as "All-Bounded" show that this number is basically $O(n \log n)$. This result implies we can save many probes if the required level of accuracy is not high.

Next we study the number of probes needed for overall estimation accuracy of 50%, 70%, 80%, and 90%. Figure 4(c) shows the accuracy of a minimum set cover (shown as "All-Bounded") for bandwidth is usually between 80% to 90%. More probes are needed for higher accuracy. This result implies the later probing packets are generally less effective than the earlier probing packets, and the probing savings is significant even for a reasonably accurate estimation.

As expected, Figure 4(a) shows the number of probing packets required for latency estimation is much higher than that of loss rate and bandwidth estimation at the same level of accuracy. This is due to the fact that a path latency estimation is a sum of the minimum latency estimation of several other paths, and thus less accurate. We show later that the improved latency estimation algorithm effectively solves this problem.

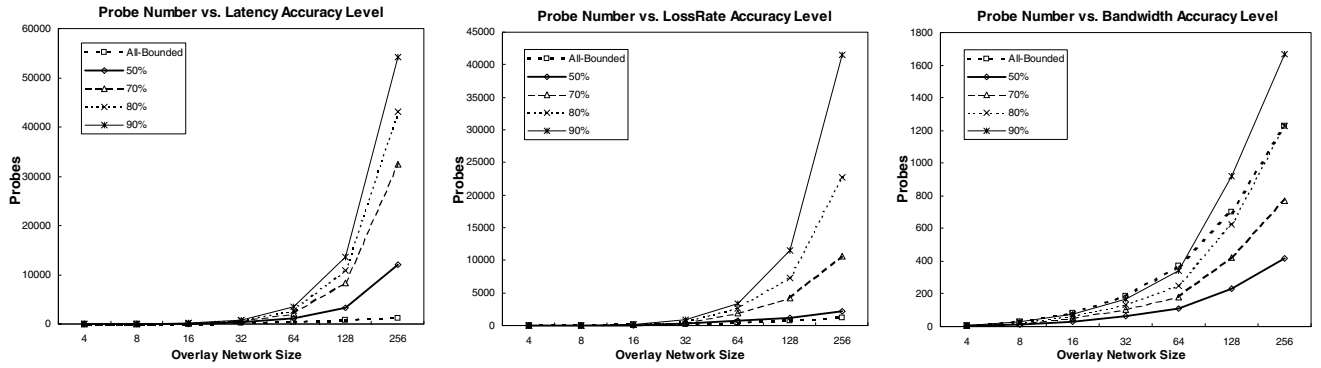
Figure 4(b) shows the performance of loss rate estimation is between bandwidth estimation and latency estimation. This is due to the fact that our loss rate estimation strategy is a combination of minimization used in bandwidth estimation, and summation used in latency estimation. We will show later the performance of the improved algorithm.

5.3. Estimation Accuracy

In this subsection we examine the estimation accuracy for a given set of probing packets. We show both the distribution and the average of the estimation accuracy of the SETCOVER_RANDOM algorithm. We provide results only for 64-node networks; additional results are in [37].

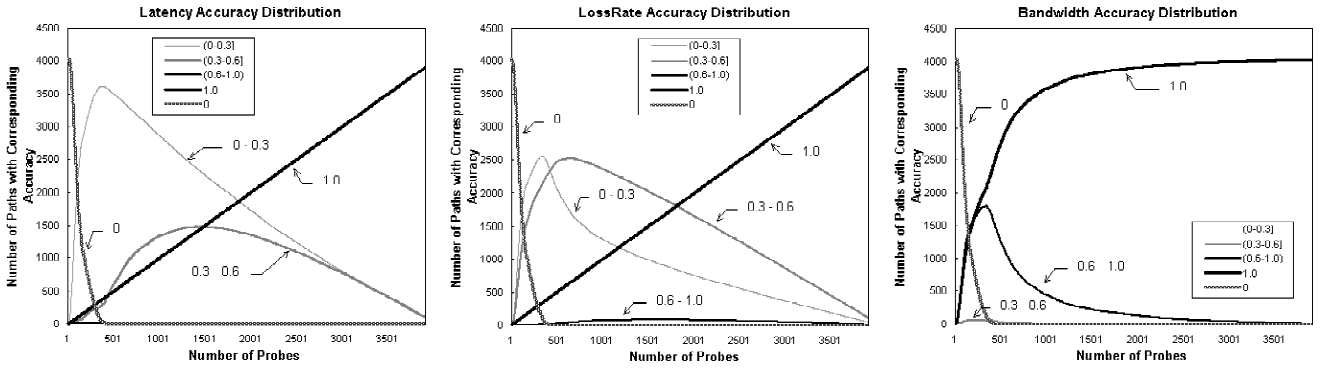
We plot the distribution of estimation accuracy versus probing packet numbers for latency, loss rate, and bandwidth in Figure 5(a), Figure 5(b), and Figure 5(c), respectively. Figure 5(c) shows the number of inaccurate estimations (those with accuracy value lower than 1.0) for bandwidth drops quickly as the probing number increases. After a relatively small number of probes, most paths can obtain exact values (accuracy value be 1.0) for bandwidth. These figures confirm the implications of the last subsection, that later probing packets are generally less useful than the earlier probing packets, and that the probing savings is significant even for a reasonably accurate estimation.

Figure 5(a) and Figure 5(b) show again the latency and loss rate estimations are less accurate than that of band-



(a) latency estimation accuracy. (b) loss rate estimation accuracy. (c) bandwidth estimation accuracy.

Figure 4. Minimum number of probes for given levels of estimation accuracy (SETCOVER_RANDOM).



(a) latency estimation accuracy. (b) loss rate estimation accuracy. (c) bandwidth estimation accuracy.

Figure 5. The estimation accuracy distribution versus number of probes (SETCOVER_RANDOM).

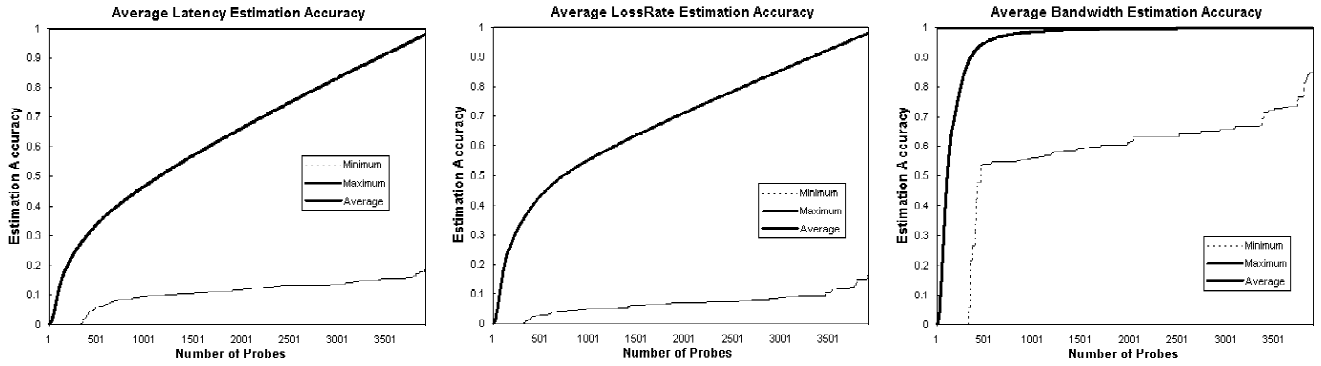
width. The number of paths with 100% accurate latency or loss rate value (accuracy value be 1.0) is exactly the same as the number of probes. This fact implies that all the ccurate latency and loss rate values are obtained by direct probing instead of by estimation, thus none of the latency estimation is 100% accurate. In addition, the figure shows a large number of paths still have low accuracy (less than 0.3) even after a substantial number of probes, which implies probe packets are less effective to enhance estimation for other paths. Both of these facts contribute to the low performance of latency and loss rate estimation.

Figure 6 plots the maximum (always 1.0), average, and minimum estimation accuracy for latency, loss rate, and bandwidth. Again, these figures show we can achieve high estimation accuracy with relatively few probes, especially in the case of bandwidth availability.

5.4. Algorithm Comparison

In this subsection we compare the performance of our path selection algorithms in terms of probing cost and estimation accuracy. We also present the performance of the improved estimation algorithms.

First we compare the algorithms NULL_RANDOM, SETCOVER_RANDOM, and WSETCOVER_RANDOM. We plot the results in Figure 7(a) and Figure 7(b). The figures show the algorithms with set cover strategies can reduce probing cost or improve estimation accuracy as expected. In addition, the weighted set cover algorithm is better than the set cover algorithm on average link stress

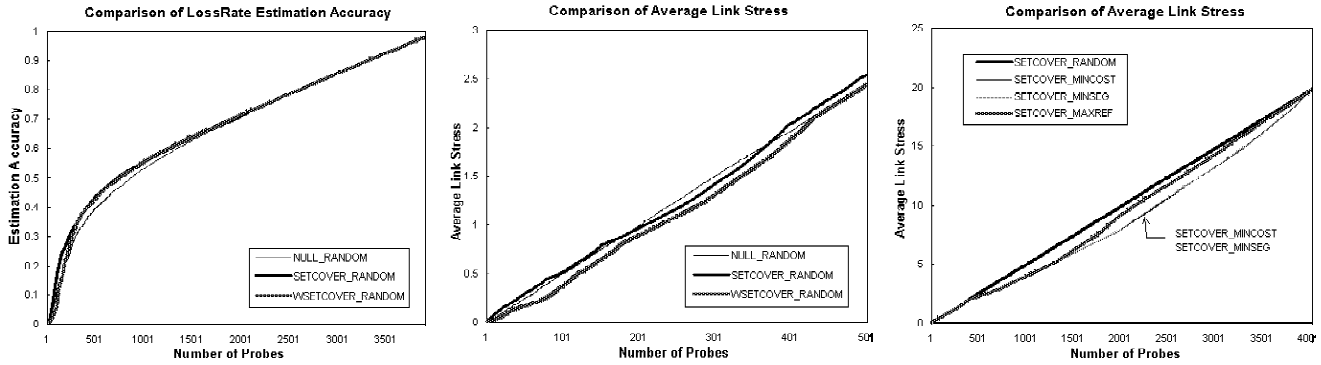


(a) latency estimation accuracy.

(b) loss rate estimation accuracy.

(c) bandwidth estimation accuracy.

Figure 6. The average estimation accuracy (SETCOVER_RANDOM).



(a) Comparison of set cover strategies on loss rate estimation.

(b) Comparison of set cover strategies on average link stress.

(c) Comparison of direct selection strategies on average link stress.

Figure 7. Comparison of path selection algorithms.

since it selects paths with low probing cost that usually corresponds to shorter paths in term of physical hops.

Next we compare four set cover algorithms: SETCOVER_RANDOM, SETCOVER_MINCOST, SETCOVER_MINSEG, and SETCOVER_MAXREF. Surprisingly these algorithms exhibit no distinguishable performance difference on path quality estimations. However, they do have different effect on average link stress. The results are shown in Figure 7(c). As expected, the algorithms of SETCOVER_MINCOST and SETCOVER_MINSEG can reduce probing cost in term of link stress. The algorithm SETCOVER_MAXREF incurs slightly higher link stress than SETCOVER_MINCOST and SETCOVER_MINSEG, but still lower than the SETCOVER_RANDOM algorithm. These performance results suggest, depending on the probing overhead definition, the algorithms SETCOVER_MINCOST/_MINSEG or WSETCOVER_MINCOST/_MINSEG are most efficient in terms of cost-accuracy ratio.

Now we consider the performance of the improved estimation algorithm. As shown in Figure 8, this algorithm can significantly improve the performance for both latency and loss rate estimation. We see there exists a threshold point, below which our original algorithm performs better than the algebraic method. After this point, the algebraic method quickly derives the quality of all paths. Our combined algorithm achieves optimity in all cases.

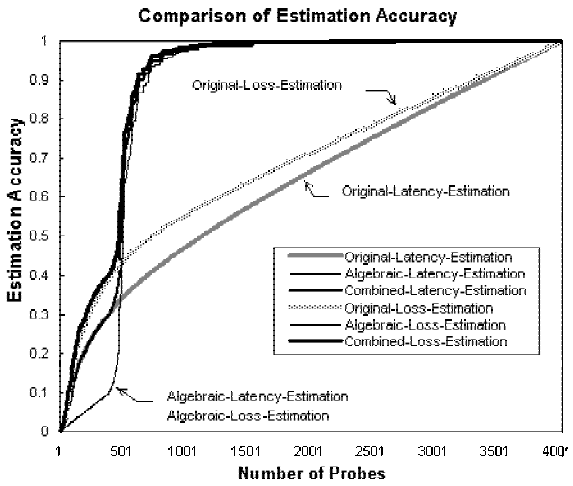


Figure 8. Improved estimation algorithm results.

5.5. Summary

Our simulation results show we can save a large amount of probing overhead while maintaining reasonably accurate quality estimation for all paths. Specifically, depending on network topology, we can obtain bounded bandwidth estimation with overall accuracy up to 90% for all paths with $O(n \log n)$ probing overhead. The improved estimation algorithm can achieve comparable performance for latency and loss rate estimations. Our results show that the later probing packets are generally less effective than the earlier probing packets in term of accuracy improvement, so it is appropriate to trade these probes for large probing cost saving. In addition, the set cover strategies are effective for estimation accuracy improvement, while other path selection strategies help reduce average link stress.

6. Discussion

The approach proposed in this paper is based on the assumption that overlay paths significantly overlap with one another. If there is little overlap, our solution will provide little or no benefit. However, in this case, the worst-case link stress is likely to be low even for a pair-wise probing. On the other hand, our simulation results show that there exists significant overlap for randomly generated overlay networks in a sparse network like the Internet.

Our solution is most suitable for applications that require only a bounded estimation for all paths. A typical example is RON [2]. The estimation bound can be very loose in RON if we only need to bypass path outages, that is, where any path with finite latency will be eligible. In that case the leader node can inform all other nodes when an alternative path is found after a path outage is detected.

Another strategy to improve the estimation accuracy is to exploit historical path quality data. Although our system frequently probes paths to guarantee high path quality, the actual quality changes on segments are not necessarily so frequent. If there exists "quality locality" then we can infer future path quality from past quality measurement or inference. We point out this inference is not bounded, however. It cannot replace our approach, but may be helpful to our algorithms. For example, we should always choose paths with low latency, low loss rate and high bandwidth to probe, so that the estimation accuracy for other paths can be higher. Specifically, since a probe on high quality path can generate high quality estimation for segments in that path, it will in turn produce high quality estimations for all other related paths. The historical data is the main reference from which we can make such a path selection.

7. Conclusions and Future Work

In this paper we proposed an overlay path probing approach that can trade probing overhead for accuracy of path quality estimation. The approach is based on the assumption that overlay paths significantly overlap with each other, enabling the probing result for one path to reveal partial quality information for other paths. We exploited this property to provide bounded estimation for unprobed paths. The quality metrics include latency, loss rate and available bandwidth. We designed several algorithms to select paths for probing. Their effects on probing cost and estimation accuracy were compared. Simulation results show that, depending on the topology, our approach can provide quality estimation with up to 90% average accuracy for all paths with $O(n \log n)$ probing overhead. We anticipate that this method can be helpful in supporting adaptive distributed systems, which require efficient network management and reconfiguration.

In the future, we plan to analyze the effect of topology information incompleteness on the performance of our estimation algorithms. The motivation is to determine the applicability of our approach on the cases where topology information is only partially available. We also plan to compare the performance of our approach to other cost-saving probing approaches. However, we emphasize that our method focuses on the cost-accuracy tradeoff instead of the cost-completeness tradeoff, and thus is orthogonal to many other approaches. In addition to theoretical analysis, we intend to implement an overlay system that employs our probing algorithms. An actual implementation may help us further evaluate the strengths and limitations of our approach in terms of performance and practicality.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable suggestions for improving the paper. This work was supported in part by the

U.S. Department of the Navy, Office of Naval Research under Grant No. N00014-01-1-0744. This work was also supported in part by U.S. National Science Foundation grants CCR-9912407, EIA-0000433, and EIA-0130724.

References

[1] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proceedings of ACM Sigmetrics*, pp. 1–12, June 2000.

[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proceedings of 18th ACM Symposium on Operating Systems Principles (SOSP)*, pp. 131–145, October 2001.

[3] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proceedings of ACM SIGCOMM*, pp. 251–262, September 1999.

[4] NLANR, National Laboratory for Applied Network Research, *http://moat.nlanr.net/Routing/rawdata*, 2000.

[5] R. Braynard, D. Kostic, A. Rodriguez, J. Chase, and A. Vahdat, "Opus: an overlay peer utility service," in *Proceedings of the 5th International Conference on Open Architectures and Network Programming (OPENARCH)*, June 2002.

[6] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *Proceedings of 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp. 329–350, November 2001.

[7] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Tech. Rep. UCB/CSD-01-1141, Computer Science Division, U.C. Berkeley, April 2001.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of ACM SIGCOMM*, pp. 161–172, August 2001.

[9] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM SIGCOMM*, pp. 205–220, August 2002.

[10] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: A framework for delivering multicast to end users," in *Proceedings of IEEE INFOCOM*, pp. 1366–1375, June 2002.

[11] A. Shaikh, M. Goyal, A. Greenberg, R. Rajan, and K. Ramakrishnan, "An OSPF topology server: Design and evaluation," *IEEE Journal of Selected Areas in Communications: Special Issue on Recent Advances on Fundamentals of Network Management*, vol. 20, pp. 746–755, May 2002.

[12] A. Bestavros, J. Byers, and K. Harfoush, "Inference and labeling of metric-induced network topologies," in *Proceedings of IEEE INFOCOM*, pp. 628–637, June 2002.

[13] M. Coates, R. Castro, and R. Nowak, "Maximum likelihood network topology identification from edge-based unicast measurements," in *Proceedings of ACM SIGMETRICS*, June 2002.

[14] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *Proceedings of ACM SIGCOMM*, August 2002.

[15] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology inference in the presence of anonymous routers," in *Proceedings of IEEE INFOCOM*, April 2003.

[16] M. Kwon and S. Fahmy, "Topology-aware overlay networks for group communication," in *Proceedings of The 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2002)*, May 2002.

[17] W. Cui, I. Stoica, and R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks," in *Proceedings of 10th IEEE International Conference on Network Protocols (ICNP'02)*, pp. 236–247, November 2002.

[18] M. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2001.

[19] N. Duffield and F. Presti, "Multicast inference of packet delay variance at interior network links," in *Proceedings of IEEE INFOCOM*, pp. 1351–1360, March 2000.

[20] F. Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," Tech. Rep. TR99-55, University of Massachustts, November 1999.

[21] V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Server-based inference of Internet link lossiness," in *Proceedings of IEEE INFOCOM*, April 2003.

[22] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," in *Proceedings of ACM SIGMETRICS*, pp. 21–30, June 2002.

[23] N. Duffield, F. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proceedings of IEEE INFOCOM*, pp. 915–923, April 2001.

[24] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM*, August 2002.

[25] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 1997.

[26] G. Liang and B. Yu, "Pseudo likelihood estimation in network tomography," in *Proceedings of IEEE INFOCOM*, April 2003.

[27] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of network topology measurements," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[28] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," in *Proceedings of IEEE INFOCOM*, April 2003.

[29] Y. Shavitt, X. Sun, A. Wool, and B. Yener, "Computing the unmeasured: an algebraic approach to Internet mapping," in *Proceedings of IEEE INFOCOM*, pp. 1646–1654, April 2001.

[30] C. Tang and P. K. McKinley, "A distributed approach to topology-aware overlay path monitoring," Tech. Rep. MSU-CSE-03-24, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, August 2003.

[31] Performance measurement tools taxonomy, *http://www.caida.org/tools/taxonomy/performance.xml*.

[32] P. Crescenzi, V. Kann, M. Halldrsson, M. Karpinski, and G. Woeginger, "A compendium of NP optimization problems," *http://www.nada.kth.se/∼viggo/wwwcompendium/wwwcompendium.html*.

[33] D. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, pp. 256–278, 1974.

[34] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[35] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of IEEE INFOCOM*, pp. 40–52, March 1996.

[36] J. Winick and S. Jamin, "Inet 3.0: Internet topology generator," Tech. Rep. CSE-TR-456-02, Department of Computer Science, University of Michigan, Ann Arbor, Michigan, 2002.

[37] C. Tang and P. K. McKinley, "On the cost-quality tradeoff in topology-aware overlay path probing," Tech. Rep. MSU-CSE-03-9, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, May 2003.